WHAT IS CLAIMED IS:

1.   A method of streaming an archive file from a server to a client device, the method comprising:

at a server, extracting ones of a plurality of modules from a first archive file;

streaming the extracted modules from the server to the client device;

receiving the streamed modules at the client device;

automatically constructing a second archive file at the client device, the second archive file comprising the received modules; and

providing data from at least one of the received modules in the second archive to an executing application.

2.   The method of claim 1 wherein:

the first archive file is a Java Archive file;

a first one of the streamed modules comprises a ZIP file end-of-central-directory record; and

the method further comprises allocating storage space at the client device for the second archive based on size information in the end-of-central-directory record.

3.   The method of claim 2 wherein the client device comprises a Java Micro Edition execution environment.

4.   The method of claim 1 wherein streaming comprises streaming in accordance with predetermined criteria predicting an order of utilization of modules at the client device.

5. The method of claim 1 wherein:

the first archive file comprises a Java Archive file;

a first one of the streamed modules comprises a Java class file;

the executing application comprises a Java application; and

5   providing data from at least one of the stored file to the executing application comprises

dynamically loading the Java class file for access by the Java application.

6. The method of claim 5 wherein:

at least one of the modules in the Java Archive file comprises Java Archive meta-

information comprising an identification of files in other ones of the modules; and

10   the method further comprises streaming the meta-information from the server to the client

device and validating a file in a received modules using the meta-information.

7. The method of claim 6 wherein validating comprises comparing a file name and path

associated with a file received in a streamed module to a file name and path specified in

the meta-information.

15   8. The method of claim 6 wherein the meta-information comprises a first digital signature

and the method further comprises:

    computing a second digital signature based on data in a first one of the received

        modules; and

    comparing the second digital signature to the first digital signature to determine

20   validity of a file received in a streamed module.

9. The method of claim 1 wherein the method further comprises:

at the client device, processing a request from the executing application to access a
file in a first one of the modules; and

determining if the first module is present in the second archive;

5   sending streaming control data to the server to request the first module when the first
module is not present in the second archive, and

streaming the first module from the server to the client device in response to receipt
of said streaming control data at the server.

10. The method of claim 9 wherein streaming the first module in response to the receipt of
10   said streaming control data comprises interrupting streaming of another one of the
modules.

11. A computer-implemented method of transmitting data in an archive file from a server
device to a client device, the client device comprising an execution environment
configured to provide ones of a collection of logically separate files in a received archive
15   ·   file to an executing application in an execution-time determined order, the method
comprising:

at the server device, extracting ones of a collection of logically separate files from a first
archive file; and

streaming the extracted ones of the separate files from the server device to the client
20   device.

12. The method of claim 11 further comprising:

receiving the streamed files at the client device; and

constructing a second archive files at the client device from the received files.

13. The method of claim 12 wherein the second archive file comprises a different arrangement of the streamed files than in the first file, said different arrangement being functionally equivalent to an arrangement of files in the first archive file.

14. The method of claim 11 wherein the first archive comprises at least one file comprising
5   non-executable data and at least one file comprising executable program code.

15. The method of claim 11 further comprising
    at the first computer, ordering the separate files extracted from the archive file based on
        criteria predicting an order of utilization of the separate files at the second computing
        device; and
10   streaming the separate files from the first computer to the second computer in accordance
        with the ordering.

16. The method of claim 15 wherein:
    the criteria comprises data stored at the first computer in a streaming control database;
        and
15   the method further comprises:
        receiving a request at the first computer from the second computer; and
        modifying the predictive data stored in the streaming control database based on the
            request.

17. The method of claim 15 wherein:
20   the criteria comprise data stored at the first computer in a streaming control database
        comprising transition records associating weighted values with execution transitions
        between selected files in the collection; and
        ordering the separate files comprises ordering based on transition record values.

18. The method of claim 17 wherein ordering based on transition record values comprises

processing transition record information using a path determination algorithm.

19. The method of claim 11 wherein streaming comprises:

receiving a first module in the module set while an application is executing;

5   storing the first module on local storage media at the second computer; and wherein

the method further comprises integrating the first module with the executing application.

20. A data storage apparatus comprising instructions to configure a computerized device to:

receive archive structure data from another computer, the archive structure data

specifying a structure of a first archive file;

10   receive from the other computer ones of the plurality of modules derived from the first

archive file;

in accordance with the archive structure data, construct a second archive file from the

received ones of the plurality of modules to produce a second archive file functionally

equivalent to the first archive file.

15 21. The data storage apparatus of claim 20 wherein:

the first archive file is a Java Archive file;

the archive structure data comprises a ZIP file central directory record; and

the instructions to construct the second archive file comprise instructions to position

the received central directory record in the second archive in accordance with ZIP

20   file format specifications.

22. The data storage apparatus of claim 21 wherein the instructions to construct in accordance with ZIP file format specifications further comprise instructions to position received ones of the modules in the second archive in an order of receipt of the modules and instructions to alter data in the second archive's central directory header to indicate

5   the placement of the ones of the modules in the second archive.

23. The data storage apparatus of claim 20 wherein the instructions further comprise instructions to:

    process a request from an executing application to access data in a first one of the modules;

10   determine if the first module is present in the second archive;

    send control data to the other computer to request the first module when the first module is not present in the second archive.

24. A data storage apparatus comprising instructions to configure a computer to:

    automatically extract a collection of logically separate modules from a first archive file;

15   and

    stream the extracted modules to a client device.

25. The data storage apparatus of claim 24 further comprising instructions to order the extracted modules based on criteria predicting an order of utilization of the modules at the client device and wherein the instructions to stream comprise instructions to stream in

20   accordance with the ordering of the modules.

26. The method of claim 25 wherein:

the instructions to order the extracted modules comprise instructions to order in

accordance with criteria retrieved from a streaming control database; and

the instructions further comprise instructions to:

5   modify the criteria stored in the streaming control database based on a module request

received from the second computer.

27. A system for transferring information modules between computers, the system

comprising:

1) a first computer, the first computer comprising:

10   a. means for executing an application,

b. means for receiving a sequence of modules associated with the application and

constructing an archive file comprising the received sequence of modules while

the application is executing, and

c. means for integrating a first module in the constructed archive file with the

15   application; and

2) a second computer, the second computer comprising:

a. means for extracting a collection of modules associated with the application from

an archive file, and

b. means for transferring the selected sequences from the second computer to the first

20   computer.

1203689_2